

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-012015

(43)Date of publication of application : 22.01.1993

(51)Int.Cl.

G06F 9/42
G06F 12/00

(21)Application number : 03-183711

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 28.06.1991

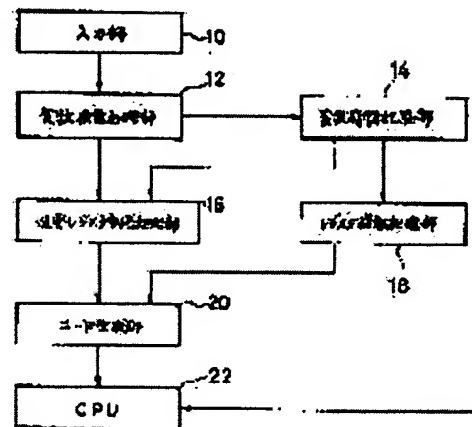
(72)Inventor : KATO MASAYA

(54) REGISTER WINDOW SETTING SYSTEM

(57)Abstract:

PURPOSE: To increase the processing execution speed by varying the window length of a register window to effectively use register resource.

CONSTITUTION: A variable retrieval processing part 12 generates the variable information of each function inputted from an input part 10. A required register assignment processing part 16 obtains the number of registers (register information of each function), which overlap at the time of calling the number of registers required in each function and the function, based on the variable information of each function. A code generating part 20 refers to register the information to generate a code including a register assigning instruction and outputs it to a CPU 22. The CPU 22 executes the actual code while controlling register windows in accordance with the register assigning instruction. Thus, required registers are assigned to each function, and the required number of registers is overlapped in the overlapping part.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-12015

(43)公開日 平成5年(1993)1月22日

(51)IntCl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/42	3 3 0 R	9189-5B		
12/00	5 9 5	8841-5B		

審査請求 未請求 請求項の数1(全 3 頁)

(21)出願番号 特願平3-183711

(22)出願日 平成3年(1991)6月28日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 加藤 雅也

東京都青梅市末広町2丁目9番地 株式会

社東芝青梅工場内

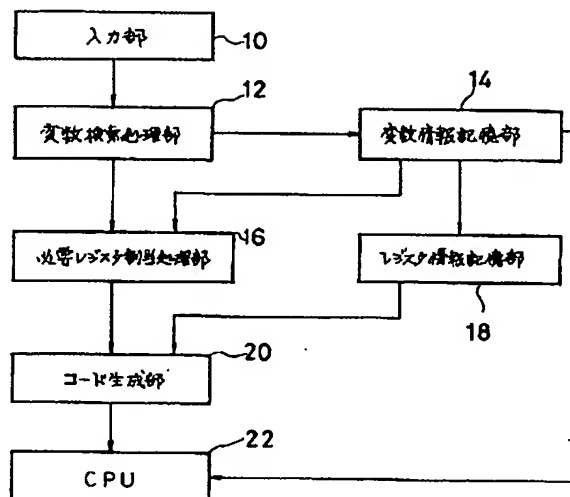
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 レジスタウィンドウ設定方式

(57)【要約】

【目的】レジスタウィンドウのウィンドウ長を可変とすることで、レジスタ資源を有効に利用し、処理の実行速度を向上させることが可能となるようにする。

【構成】入力部10から入力された関数について、変数検索処理部12は、関数毎の変数情報を生成する。必要レジスタ割当処理部16は、関数毎の変数情報に基づいて、各関数内で必要なレジスタ数と関数を呼出すときにオーバーラップさせるレジスタウィンドウの個数(関数毎のレジスタ情報)を求める。コード生成部20は、レジスタ情報を参照してレジスタ割当命令を含むコードを生成し、CPU22に出力する。CPU22は、レジスタ割当命令に従って、レジスタウィンドウのコントロールを行ないながら、実際のコードを実行する。こうして、各関数に必要なだけのレジスタが割り当てられ、またオーバーラップする部分も必要な個数だけのレジスタがオーバーラップされる。



【特許請求の範囲】

【請求項1】 オーバーラッピングレジスタウィンドウを備えた制御ユニットにおいて、レジスタウィンドウのウィンドウ長を動的に変更する可変手段を具備し、前記可変手段によって得られたレジスタウィンドウを制御することを特徴とするレジスタウィンドウ設定方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、オーバーラッピングレジスタウィンドウを備えた制御ユニットにおけるレジスタウィンドウ設定方式に関する。

【0002】

【従来の技術】 従来、オーバーラッピングレジスタウィンドウを備えた制御ユニットでは、レジスタウィンドウが固定長となっている。この従来のレジスタウィンドウの概念図を図3に示している。

【0003】 図3においては、関数fが関数gを呼び、関数gが関数hを呼ぶものとする。オーバーラッピングレジスタウィンドウでは、各々のレジスタのうち、オーバーラップする部分を用いて引数や返却値の受け渡しが行なわれる。

【0004】 従って、例えば図3に示すように、関数gがレジスタウィンドウで与えられるだけのレジスタを必要としない場合には、図中斜線部分のレジスタが無駄になってしまう。

【0005】 とところで、関数やサブルーチン呼出しが一定の深さに達し、レジスタを使い切った場合には、通常、レジスタセーブが行なわれる。また、一つの関数で沢山のレジスタを使う場合にも、同様にレジスタセーブが行なわれる。

【0006】

【発明が解決しようとする課題】 このように、従来のオーバーラッピングレジスタウィンドウは固定長であったために、変数をあまり使用しない関数やサブルーチンのように、レジスタウィンドウで与えられるだけのレジスタを必要としない場合、図3に示すように、レジスタが無駄となることがあった。このように、不必要にレジスタを浪費することは、必要以上のレジスタセーブを招くことになる。すなわち、従来では、レジスタ資源を有効に利用できず、これに伴って実行速度の低下を招いてしまうという問題があった。

【0007】 本発明は前記のような点に鑑みてなされたもので、レジスタ資源を有効に利用して、処理の実行速度を向上させることが可能なレジスタウィンドウ設定方式に関する。

【0008】

【課題を解決するための手段】 本発明は、オーバーラッピングレジスタウィンドウを備えた制御ユニットにおいて、レジスタウィンドウのウィンドウ長を動的に変更す

る可変手段を具備し、前記可変手段によって得られたレジスタウィンドウを制御するものである。

【0009】

【作用】 これにより、レジスタ資源を有効に利用でき、従って必要以上にレジスタセーブを招くことがないため、実行速度を向上させることができる。

【0010】

【実施例】 以下、図面を参照して本発明の一実施例を説明する。図1は同実施例に係わるレジスタウィンドウ設定方式を説明するための機能ブロック図である。図1において、入力部10は、処理対象とする関数を入力するものである。変数検索処理部12は、入力部10から入力された各関数が使う変数の個数や変数の種類等を求めるものである。変数情報記憶部14は、関数毎の変数情報を記憶するものであり、同時に生存する変数の個数や種類を示す。必要レジスタ割当処理部16は、変数情報記憶部14を参照して、レジスタに割り当てるべき変数を決め、関数内毎で使うレジスタの個数を求めるものである。レジスタ情報記憶部18は、関数毎のレジスタ情報を記憶するものであり、各関数が関数内で必要とするレジスタの個数及び呼出し側のレジスタウィンドウとオーバーラップするレジスタの個数を示す。コード生成部20は、関数毎のレジスタ情報を参照して、レジスタ割当命令を含むコードを生成するものである。CPU22は、コード生成部20によって生成されたコードを実行するもので、レジスタ割当命令に応じてレジスタウィンドウのコントロールを行なう。

【0011】 次に、同実施例の動作について説明する。

【0012】 入力部10から入力された関数について、変数検索処理部12は、通常のコンパイラ等が行なうような変数検索処理を関数単位で行ない、変数のサイズ、生存範囲と共に、変数の個数や種類等を含む関数毎の変数情報を生成する。ここで生成された関数毎の変数情報は、変数情報記憶部14に記憶される。

【0013】 次に、必要レジスタ割当処理部16は、変数情報記憶部14に記憶された関数毎の変数情報に基づいて、各関数内で必要なレジスタ数と、その関数を呼出すときにオーバーラップさせるレジスタウィンドウの個数、変数にレジスタを割り付けながら求める。ここで得られた関数毎のレジスタ情報は、レジスタ情報記憶部18に記憶される。

【0014】 その後、コード生成部20は、コード生成を行なうが、関数の呼出しに伴い、呼出される関数側でのレジスタウィンドウをどのような状態にするのか、レジスタ情報記憶部18に記憶された関数毎のレジスタ情報を参照して判別する。コード生成部20は、この判別結果に基づいて、レジスタ割当命令を含むコードを生成し、CPU22に出力する。

【0015】 CPU22は、レジスタ割当命令に従って、レジスタウィンドウのコントロールを行ないなが

ら、実際のコードを実行する。このCPU22によるレジスタウィンドウのコントロール概念を図2に示している。

【0016】ここでは、図3を用いて説明した際と同様の状況であり、関数fが関数gを呼び、関数gが関数hを呼ぶものとする。図2に示すように、関数gのレジスタウィンドウが、図3に示すレジスタウィンドウと異なり、各関数に必要なだけのレジスタが割り当てられている。オーバーラップする部分も同様で、関数gが関数hを呼出す場合、必要な個数だけのレジスタがオーバーラップされている。

【0017】このように、無駄となるレジスタがないため、不必要にレジスタを浪費することがない。従って、必要以上のレジスタセーブを招くことがないため、より効率的な実行が可能となるものである。

【0018】なお、レジスタが沢山ある（容量が大きい）CPUでは、ハードウェアによって可変個のレジスタウィンドウを実現しなくても、ソフトウェアによって

実現可能である。

【0019】

【発明の効果】以上のように本発明によれば、オーバーラッピングレジスタウィンドウを可変長にすることにより、レジスタをあまり必要としない関数やサブルーチンの呼出しを、より深くまでレジスタセーブすることなく呼び出すことが可能となるので、より効率的に処理を実行することが可能となるものである。

【図面の簡単な説明】

【図1】本発明の一実施例に係わるレジスタウィンドウ設定方式を説明するための機能ブロック図。

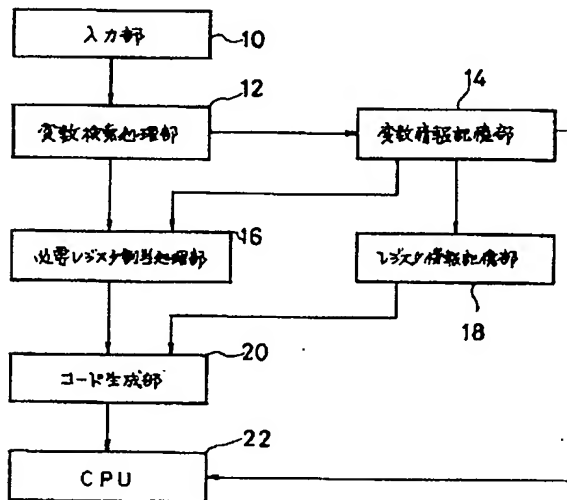
【図2】同実施例におけるレジスタウィンドウの概念図。

【図3】従来におけるレジスタウィンドウの概念図。

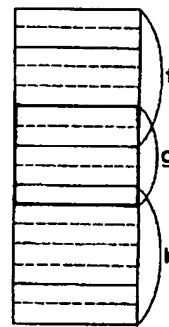
【符号の説明】

10…入力部、12…変数検索処理部、14…変数情報記憶部、16…必要レジスタ割当処理部、18…レジスタ情報記憶部、20…コード生成部、22…CPU。

【図1】



【図2】



【図3】

